

# 在 CentOS7 上的编译安装 Open vSwitch

## 目录

在 CentOS7 上的编译安装 Open vSwitch .....	5
1 准备 .....	6
1.1 源码获取 .....	6
1.2 构建依赖项 .....	6
1.3 安装的依赖项 .....	6
2 安装 .....	7
2.1 配置 .....	7
2.1.1 configure 构建选项的解释 .....	7
2.2 安装过程 .....	9
2.2.1 安装依赖包 .....	9
2.2.2 检查配置 .....	9
2.2.3 编译 .....	10
2.3 手动启动程序 .....	11
2.3.1 配置数据库 .....	11
2.3.2 配置 ovssdb-server .....	11
2.3.3 初始化数据库 .....	12
2.3.4 启动主守护程序 .....	12
2.3.5 查看 OVS 进程是否启动 .....	12
2.4 使用脚本和 ovs-ctl 的方式启动 .....	13
2.4.1 以默认方式启动/停止 .....	13
2.4.2 仅启动 ovssdb-server .....	13
2.4.3 仅启动 ovs-vswitchd .....	14
2.4.4 查看 ovs 进程是否启动 .....	14
2.5 命令使用 .....	14
2.6 升级 .....	14

## 参考文章

- [1] [Open vSwitch on Linux, FreeBSD and NetBSD](#) 参考主要编译安装过程，未采用 docker 安装方式
- [2] [How to Install and use Open vSwitch with KVM on CentOS 7 / RHEL 7](#) 参考所需依赖项，及配置错误的处理
- [3] [Centos 安装 OpenVSwitch 历程](#) 同[2]类似，从源码打包的方式安装 ovs，本篇未采用打包方式安装，仅作参考

## 安装环境

软件：CentOS 7.6

以下准备及安装过程在虚拟机下进行，若想快速体验可从 [2.2 小节 安装过程](#) 开始，仅作参考。

# 1 准备

## 1.1 源码获取

官网发布 <http://www.openvswitch.org/download/>

GitHub 上发布 <https://github.com/openvswitch/ovs/releases>

## 1.2 构建依赖项

- GNU make
- C 编译器
  - GCC 4.6 或以上 或
  - Clang 3.4 或以上 或
  - MSVC 2013（在 windows 上安装时需要）
- Libssl(可选)如果打算将 OVS 连接到 OpenFlow 控制器,建议安装 libssl(来自 OpenSSL)
- Libcap-ng (可选) 如果想支持以非 root 用户运行 OVS 守护程序, 建议安装 libcap-ng。
- Python 3.4 或以上
- Unbound 库(可选),如果要在指定 OpenFlow 和 OVSDDB 远程服务器时启用 ovs-vswitchd 和其他实用程序使用 DNS 名称, 则建议使用。环境变量 OVS\_RESOLV\_CONF 可用于指定 DNS 服务器配置文件 (Linux 上的默认文件是/etc/resolv.conf)。
- 受支持的 Linux 内核版本
  - 为支持的 ingress policing, 需要启用内核配置选项 NET\_CLS\_BASIC, NET\_SCH\_INGRESS 以及 NET\_ACT\_POLICE。(NET\_CLS\_POLICE 已过时, 不需要)
  - 在 3.11 之前的内核上, 不得加载或编译 ip\_gre 用于 IP (NET\_IPGRE) 上 GRE 隧道的模块
  - 要使用 Open vSwitch 配置 HTB 或 HFSC 服务质量, 必须启用相应的配置选项
  - 要对 TAP 设备使用 Open vSwitch 支持, 必须启用 CONFIG\_TUN
- 如果使用的是 Git 树或快照 (而不是分发 tarball), 或者修改了 Open vSwitch 构建系统或数据库架构, 则还需要以下软件:
  - Autoconf 版本 2.63 或更高版本
  - Automake 版本 1.10 或更高版本
  - libtool 2.4 或更高版本。
- 用户空间和 Linux 数据路径的数据路径测试还取决于
  - Pyftplib v1.2.0
  - GNU wget v1.16
  - Netcat
  - Curl v7.47.0
  - Tftpy v0.6.2
- 如果要大量修改 Open vSwitch, 考虑安装以下包以获得更好的警告提示
  - Sparse v0.5.1 及以上(<https://git.kernel.org/pub/scm/devel/sparse/sparse.git/>)
  - GNU make
  - Clang v3.4 及以上
  - Flake8 插件 (python 代码插件)

## 1.3 安装的依赖项

简单安装并运行 OVS 应具备的软件包：

- 与用于构建的共享库兼容
- 对于 Linux 上的入口策略的可选支持，请使用 iproute2 中的“tc”程序（可从 <https://wiki.linuxfoundation.org/networking/iproute2> 获得）
- Python v3.4 及以上或 python2.7.5 及以上

在 Linux 上，要确保该目录/dev/urandom 存在。要支持 TAP 设备，还必须确保该设备/dev/net/tun 存在。

## 2 安装

### 2.1 配置

如果是从 open vswitch 的 git 源获取的源代码，那么需要在根目录中运行 boot.sh 来构建配置脚本：

```
$ ./boot.sh
```

#### 2.1.1 configure 构建选项的解释

根据需要自行选择合适的配置选项。

默认情况下，所有文件都安装在下`/usr/local`。默认情况下，Open vSwitch 还希望在`/usr/local/etc/openvswitch`中找到其数据库。如果要将所有文件安装到例如`/usr`和`/var`中而不是`/usr/local`和`/usr/local/var`并希望`/etc/openvswitch`用作默认数据库目录，请添加如下所示的选项：

```
$ ./configure --prefix=/usr --localstatedir=/var --sysconfdir=/etc
```

- 默认情况下，将构建并链接静态库。如果要改为使用共享库：

```
$ ./configure --enable-shared
```

- 要使用特定的 C 编译器来编译 Open vSwitch 用户程序，如：

```
$ ./configure CC=gcc-4.2
```

要使用“clang”编译器：

```
$ ./configure CC=clang
```

- 要向 C 编译器提供特殊标志，请`CFLAGS`在 configure 命令行上指定它们。如果要使用默认的 CFLAGS，其中包括`-g`构建调试符号和`-O2`启用优化的功能，则必须自己包含它们。例如，要使用默认的 CFLAGS plus 进行构建`-mssse3`，可以按以下方式运行 configure：

```
$ ./configure CFLAGS="-g -O2 -mssse3"
```

- 为了进行有效的哈希计算，可以传递特殊标志以利用内置的内在函数。例如，在支持 SSE4.2 指令集的 X86\_64 上，可以通过传递 CRC32 内部函数来使用`-msse4.2`：

```
$ ./configure CFLAGS="-g -O2 -msse4.2"
```

- 内置的 `popcnt` 指令也可用于加快对整数中设置的位的计数。例如，在具有 POPCNT 支持的 X86\_64 上，可以通过以下方式启用它 `-mpopcnt`：

```
$ ./configure CFLAGS="-g -O2 -mpopcnt"
```

- 如果您使用的是其他处理器，并且不知道选择哪些标志，建议使用 `-march=native` 设置：

```
$ ./configure CFLAGS="-g -O2 -march=native"
```

这样，GCC 将检测处理器并自动为其设置适当的标志。如果要在目标计算机外部编译 OVS，则不应使用此方法。

- 如果想要启用 Address Sanitizer 进行调试，且运行时成本约为 2 倍，则可以添加到 CFLAGS。例如：`-fsanitize=address -fno-omit-frame-pointer -fno-common`：

```
$ ./configure CFLAGS="-g -O2 -fsanitize=address -fno-omit-frame-pointer -fno-common"
```

- 如果要构建 Linux 内核模块，以便可以运行基于内核的开关，则可以 `--with-linux` 方式传递内核构建目录的位置。例如，要为正在运行的 Linux 实例构建：

```
$ ./configure --with-linux=/lib/modules/$(uname -r)/build
```

- 如果要为用于构建的机器之外的体系结构构建内核模块，则可以在调用配置脚本时使用 KARCH 变量指定内核体系结构字符串。例如，要在 Linux 上为 MIPS 构建：

```
$ ./configure --with-linux=/path/to/linux KARCH=mips
```

- 如果计划进行很多 Open vSwitch 开发，则可能需要添加 `--enable-Werror`，这会将 `-Werror` 选项添加到编译器命令行中，从而将警告变为错误。这样就不可能错过构建生成的警告。例如：

```
$ ./configure --enable-Werror
```

- 如果要使用 GCC 进行构建，并且为了改进警告，可以安装 `sparse`（请参阅“先决条件”）并通过添加启用它以进行构建 `--enable-sparse`。将此与 `--enable-Werror` 一起使用可避免同时丢失编译器和 `sparse` 警告，例如：

```
$ ./configure --enable-Werror --enable-sparse
```

- 如果要使用 gcov 代码覆盖率支持进行构建，请添加 `--enable-coverage`：

```
$ ./configure --enable-coverage
```

- 使用 `--help` 查看可调用的 configure 选项：

```
$ ./configure --help
```

- 从单独的构建目录运行 configure。如果要从单个源目录以多种方式构建 Open vSwitch，例如尝试 GCC 和 Clang 构建，或为多个 Linux 版本构建内核模块。例如：

```
$ mkdir _gcc && (cd _gcc && ./configure CC=gcc)
$ mkdir _clang && (cd _clang && ./configure CC=clang)
```

- 在某些负载下，使用 jemalloc 内存分配器而不是 glibc 内存分配器时，ovsdb 服务器和其他组件的性能会更好。如果您希望与 jemalloc 链接，请将其添加到 LIBS：

```
$ ./configure LIBS=-ljemalloc
```

## 2.2 安装过程

### 2.2.1 安装依赖包

```
$ yum install -y wget gcc make python-devel openssl-devel kernel-devel graphviz kernel-debug-devel autoconf automake rpm-build redhat-rpm-config libtool
```

```
$ yum install -y '@Development Tools' rpm-build yum-utils
```

```
$ yum install -y libcap-ng
```

### 2.2.2 检查配置

- 下载安装包并解压，比如下载官网提供的最新 2.9.x 版本：

```
$ wget https://www.openvswitch.org/releases/openvswitch-2.9.5.tar.gz
```

- 解压

```
$ tar -zxf openvswitch-2.9.5.tar.gz
```

- 在文件根目录下执行配置文件

```
$ ./configure
```

看到以下信息：

```
[root@keydb1 openvswitch-2.9.5]# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
...
...
configure: WARNING: cannot find libcap-ng.
--user option will not be supported on Linux.
(you may use --disable-libcapng to suppress this warning).
checking for Python 2.x for x >= 7... /usr/bin/python
checking whether /usr/bin/python has six library... configure: error: Missing Python six library or
version too old.
```

- (a) libcap-ng 未找到，但是在前述步骤中显示系统已安装有 libcap-ng，这里建议使用配置选项 `--disable-libcapng`

(b) python six 库的问题，通过安装 python-six 包解决

```
$ yum install -y python-six
```

再执行 `./configure` 无错误产生。

### 2.2.3 编译

- 在文件根目录下执行编译

```
$ make

...

...

./build-aux/cksum-schema-check ovn/ovn-nb.ovsschema ovn/ovn-nb.ovsschema.stamp
./build-aux/cksum-schema-check ovn/ovn-sb.ovsschema ovn/ovn-sb.ovsschema.stamp
make[2]: Leaving directory `/usr/local/openvswitch-2.9.5'
make[1]: Leaving directory `/usr/local/openvswitch-2.9.5'
```

- 执行编译检查（可选）

```
$ make check

...

...

## ----- ##
## Test results. ##
## ----- ##

2225 tests were successful.
322 tests were skipped.
make[3]: Leaving directory `/usr/local/openvswitch-2.9.5'
make[2]: Leaving directory `/usr/local/openvswitch-2.9.5'
make[1]: Leaving directory `/usr/local/openvswitch-2.9.5'
```

- 安装

```
$ make install
make install-recursive
make[1]: Entering directory `/usr/local/openvswitch-2.9.5'
Making install in datapath
make[2]: Entering directory `/usr/local/openvswitch-2.9.5/datapath'
make[3]: Entering directory `/usr/local/openvswitch-2.9.5/datapath'
...
...
make[3]: Leaving directory `/usr/local/openvswitch-2.9.5'
```

```
make[2]: Leaving directory `/usr/local/openvswitch-2.9.5'
make[1]: Leaving directory `/usr/local/openvswitch-2.9.5'
```

- 检查内核模块

如果在安装的过程中生成了修改了内核模块，那么重新编译内核：

```
[root@keydb1 openvswitch-2.9.5]# make modules_install
make: Nothing to be done for `modules_install'.
```

- 载入模块并校验

载入 openvswitch 的模块到内核中：

```
[root@keydb1 openvswitch-2.9.5]# /sbin/modprobe openvswitch
[root@keydb1 openvswitch-2.9.5]# /sbin/lsmmod | grep openvswitch
openvswitch                114793    0
nf_defrag_ipv6             35104    2 openvswitch,nf_conntrack_ipv6
nf_nat_ipv6                14131    2 openvswitch,ip6table_nat
nf_nat_ipv4                14115    2 openvswitch,ip6table_nat
nf_nat                     26787    3 openvswitch,nf_nat_ipv4,nf_nat_ipv6
nf_conntrack               133095    8
ip_vs,openvswitch,nf_nat,nf_nat_ipv4,nf_nat_ipv6,xt_conntrack,nf_conntrack_ipv4,nf_conntrack_ipv6
libcrc32c                  12644    5 xfs,ip_vs,openvswitch,nf_nat,nf_conntrack
```

## 2.3 手动启动程序

### 2.3.1 配置数据库

在启动 ovs-vswitchd 本身之前，需要启动其配置数据库 ovssdb-server。安装了 Open vSwitch 的每台计算机都应运行自己的 ovssdb-server 副本。在启动 ovssdb-server 本身之前，配置可以使用的数据库：

```
$ mkdir -p /usr/local/etc/openvswitch
$ ovssdb-tool create /usr/local/etc/openvswitch/conf.db \
    vswitchd/vswitch.ovsschema
```

### 2.3.2 配置 ovssdb-server

配置 ovssdb-server 以使用上面创建的数据库，侦听 Unix 域套接字，连接到数据库本身中指定的任何管理器，以及使用数据库中的 SSL 配置：

```
$ mkdir -p /usr/local/var/{run,log}/openvswitch
$ ovssdb-server --remote=punix:/usr/local/var/run/openvswitch/db.sock \
    --remote=db:Open_vSwitch,Open_vSwitch,manager_options \
    --private-key=db:Open_vSwitch,SSL,private_key \
    --certificate=db:Open_vSwitch,SSL,certificate \
```

```
--bootstrap-ca-cert=db:Open_vSwitch,SSL,ca_cert \  
--pidfile --detach --log-file  
2020-03-11T03:24:40Z|00001|vlog|INFO|opened log file /usr/local/var/log/openvswitch/ovsdb-  
server.log  
ovsdb-server: /usr/local/var/run/openvswitch/ovsdb-server.pid: already running as pid 66781,  
aborting
```

**注意：**

如果创建的 Open vSwitch 不支持 SSL，那么应该省略--private-key， --certificate 和--bootstrap-ca-cert)

### 2.3.3 初始化数据库

使用 ovs-vsctl 初始化数据库。尽管在任何时候都可以运行它，但这在使用 ovsdb-tool 创建数据库后必须要执行一次初始化：

```
$ ovs-vsctl --no-wait init
```

### 2.3.4 启动主守护程序

启动主 Open vSwitch 守护程序，告诉它连接到相同的 Unix 域套接字：

```
$ ovs-vswitchd --pidfile --detach --log-file  
  
[root@keydb1 openvswitch]# ovs-vswitchd --pidfile --detach --log-file  
2020-03-11T03:20:00Z|00001|vlog|WARN|opened log file /usr/local/var/log/openvswitch/ovs-  
vswitchd.log  
2020-03-11T03:20:00Z|00002|ovs_numa|INFO|Discovered 4 CPU cores on NUMA node 0  
2020-03-11T03:20:00Z|00003|ovs_numa|INFO|Discovered 1 NUMA nodes and 4 CPU cores  
2020-03-11T03:20:00Z|00004|reconnect|INFO|unix:/usr/local/var/run/openvswitch/db.sock:  
connecting...  
2020-03-11T03:20:00Z|00005|reconnect|INFO|unix:/usr/local/var/run/openvswitch/db.sock:  
connected
```

### 2.3.5 查看 OVS 进程是否启动

```
[root@keydb1 var]# ps -ef | grep ovs  
root      66781      1  0 11:05 ?          00:00:01 ovsdb-server -v --  
remote=punix:/usr/local/var/run/openvswitch/db.sock --  
remote=db:Open_vSwitch,Open_vSwitch,manager_options --private-  
key=db:Open_vSwitch,SSL,private_key --certificate=db:Open_vSwitch,SSL,certificate --  
bootstrap-ca-cert=db:Open_vSwitch,SSL,ca_cert --pidfile --detach --log-file  
root      66818      1  0 11:20 ?          00:00:00 ovs-vswitchd --pidfile --detach --log-file  
root      67135   19121  0 13:13 pts/0    00:00:00 grep --color=auto ovs
```



查看 ovs 版本号

```
[root@keydb1 var]# ovs-vsctl --version
ovs-vsctl (Open vSwitch) 2.9.5
DB Schema 7.15.1
```

## 2.4 使用脚本和 ovs-ctl 的方式启动

Open vSwitch 包含一个外壳脚本和一个名为 ovs-ctl 的帮助程序，该程序可自动启动和停止 ovssdb-server 和 ovs-vswitchd 的许多任务。安装后，可以使用 ovs-ctl 实用程序启动守护程序。为它设置初始条件，它将以正确的顺序启动守护程序。ovs-ctl 实用程序位于 '\$ {pkgdatadir}/scripts' 中，默认为 '/usr/local/share/openvswitch/scripts'。示例：

```
$ export PATH=$PATH:/usr/local/share/openvswitch/scripts
$ ovs-ctl [options] commands
```

Commands:

start	start Open vSwitch daemons
stop	stop Open vSwitch daemons
restart	stop and start Open vSwitch daemons
status	check whether Open vSwitch daemons are running
version	print versions of Open vSwitch daemons
load-kmod	insert modules if not already present
force-reload-kmod	save OVS network device state, stop OVS, unload kernel module, reload kernel module, start OVS, restore state
enable-protocol	enable protocol specified in options with iptables
delete-transient-ports	delete transient (other_config:transient=true) ports
help	display this help message

### 2.4.1 以默认方式启动/停止

```
Starting ovssdb-server [ OK ]
system ID not configured, please use --system-id ... failed!
Configuring Open vSwitch system IDs [ OK ]
Starting ovs-vswitchd [ OK ]
Enabling remote OVSDB managers [ OK ]
```

ovs-ctl 脚本允许使用特定选项分别启动/停止守护程序，更更多的选项可查看 **ovs-ctl --help**。

### 2.4.2 仅启动 ovssdb-server

```
$ export PATH=$PATH:/usr/local/share/openvswitch/scripts
$ ovs-ctl --no-ovs-vswitchd start
```

### 2.4.3 仅启动 ovs-vswitchd

```
$ export PATH=$PATH:/usr/local/share/openvswitch/scripts
$ ovs-ctl --no-ovsdb-server start
```

### 2.4.4 查看 ovs 进程是否启动

```
[root@keydb2 openvswitch-2.9.5]# ps -ef | grep ovs
root 70687 1 0 13:03 ? 00:00:00 ovsdb-server: monitoring pid 70688 (healthy)
root 70688 70687 0 13:03 ? 00:00:00 ovsdb-server /usr/local/etc/openvswitch/conf.db -
vconsole:emer -vsyslog:err -vfile:info --remote=punix:/usr/local/var/run/openvswitch/db.sock --
private-key=db:Open_vSwitch,SSL,private_key --certificate=db:Open_vSwitch,SSL,certificate --
bootstrap-ca-cert=db:Open_vSwitch,SSL,ca_cert --no-chdir --log-
file=/usr/local/var/log/openvswitch/ovsdb-server.log --
pidfile=/usr/local/var/run/openvswitch/ovsdb-server.pid --detach --monitor
root 70696 1 0 13:03 ? 00:00:00 ovs-vswitchd: monitoring pid 70697 (healthy)
root 70697 70696 0 13:03 ? 00:00:00 ovs-vswitchd unix:/usr/local/var/run/openvswitch/db.sock -
vconsole:emer -vsyslog:err -vfile:info --mlockall --no-chdir --log-
file=/usr/local/var/log/openvswitch/ovs-vswitchd.log --pidfile=/usr/local/var/run/openvswitch/ovs-
vswitchd.pid --detach --monitor
root 70758 108538 0 13:15 pts/1 00:00:00 grep --color=auto ovs
```

查看 ovs 版本号

```
[root@keydb2 openvswitch-2.9.5]# ovs-ctl version
ovsdb-server (Open vSwitch) 2.9.5
ovs-vswitchd (Open vSwitch) 2.9.5
```

或者

```
[root@keydb2 openvswitch-2.9.5]# ovs-vsctl --version
ovs-vsctl (Open vSwitch) 2.9.5
DB Schema 7.15.1
```

## 2.5 命令使用

可参考：

[OpenvSwitch 系列之三 ovs-vsctl 命令使用](#)

[Open vSwitch 2.3.0 版本安装部署及基本操作](#) 可参考其中如何连接控制器的步骤，在此引用备忘

## 2.6 升级

未测试，可参考

<http://docs.openvswitch.org/en/stable/intro/install/general/> 提供一般升级与热升级方式